WHAT IS CLAIMED IS:

1. An apparatus for quality-of-service-controllable real-time scheduling, the apparatus comprising:

a regulator for receiving a plurality of tasks for the apparatus;

an on-line scheduler, coupled to the regulator, for selecting a real-time scheduling method and receiving a number of the tasks, wherein the number of the tasks which are inputted to the on-line scheduler are adjusted by the regulator, and the on-line scheduler, according to the real-time scheduling method, is to configure time intervals for inputted tasks to be executed; and

an evaluator, coupled to the regulator and the on-line scheduler, for evaluating a scheduling result of the on-line scheduler, feeding a first set of parameters into the regulator for a coarse adjustment, and feeding a second set of parameters into the on-line scheduler for a fine adjustment.

2. An apparatus according to claim 1, wherein the regulator, the on-line scheduler, and the evaluator are hardware devices.

3. An apparatus according to claim 1, wherein each of the tasks includes a mandatory portion and an optional portion, and the fine adjustment is performed by controlling a proportion of the optional portion to be executed.

4. An apparatus according to claim 1 or 3, wherein the regulator adjusts the

number of the tasks inputted to the on-line scheduler according to the first set of parameters.

5. An apparatus according to claim 1 or 3, wherein the on-line scheduler selects the real-time scheduling method according to the second set of parameters.

6. An apparatus according to claim 5, wherein the real-time scheduling method is a scheduling method in which the mandatory portions of the inputted tasks are executed as soon as possible and the optional portions of the inputted tasks are substitutable (MOS method).

7. An apparatus according to claim 5, wherein the real-time scheduling method is a scheduling method in which the mandatory portions of the inputted tasks are executed as soon as possible and the substitutable optional portions of the inputted tasks are postponed (MOP method).

8. An apparatus according to claim 5, wherein the real-time scheduling method is a scheduling method in which the mandatory portions of the inputted tasks are executed as soon as possible and the optional portions of the inputted tasks are to be executed fairly (MOF method).

9. An apparatus according to claim 1 or 3, wherein the evaluator evaluates the scheduling result of the on-line scheduler according an evaluation standard, and the evaluation standard includes a task rejection rate, a task suspend/discard rate, an idle rate, and a slack time.

10. An apparatus according to claim 1 or 3, wherein the first set of parameters includes a token generation rate.

11. An apparatus according to claim 1 or 3, wherein the first set of parameters includes a token number.

12. An apparatus according to claim 1 or 3, wherein the first set of parameters includes a queue length.

13. An apparatus according to claim 1 or 3, wherein the second set of parameters includes a real-time scheduling selection parameter.

14. An apparatus according to claim 1 or 3, wherein the second set of parameters includes a substitutable check parameter.

15. An apparatus according to claim 1 or 3, wherein the second set of parameters includes a parameter indicative of a maximum allowable execution proportion of the optional portion.

16. In a real-time scheduling mechanism for scheduling a plurality of tasks T, a method for 1-task-look-ahead substitutable checking, wherein the plurality of tasks T include respective mandatory portions M and optional portions O, the mandatory portions M of the plurality of tasks T are scheduled in a reservation list according to a model so that each of the mandatory portions M_i scheduled in the reservation list has a starting time s_i and an ending time f_i, and after the mandatory portion M_i of each of the tasks T_i is completed, the method is performed on the task T_i, wherein, for each of

the tasks T_i, the mandatory portion M_i has a processing time m_i, the optional portion O_i has a processing time o_i, and the subscript *i* indexes a sequence of starting time for the mandatory portions M scheduled in the reservation list, the method comprising the steps of:

(a) determining an effective interval t_i according to the starting time s_{i+1} of the mandatory portion M_{i+1} of a task T_{i+1}, the ending time f_i of the mandatory portion M_i of the task T_i, and an interval for the processing time o_i of the optional portion O_i of the task T_i;

(b) moving the mandatory portion M_{i+1} to a location in the reservation list by setting the starting time s_{i+1} of the mandatory portion M_{i+1} to be the ending time f_i of the mandatory portion M_i and by changing the ending time f_{i+1} of the mandatory portion M_{i+1} correspondingly;

(c) determining an effective interval t_{i+1} according to the starting time s_{i+2} of the mandatory portion M_{i+2} of a task T_{i+2}, the ending time f_{i+1} of the mandatory portion M_{i+1}, and an interval for the processing time o_{i+1} of the optional portion O_{i+1} of the task T_{i+1}; and

(d) comparing the effective interval t_i with the effective interval t_{i+1}; if the effective interval t_i is less than the effective interval t_{i+1}, the optional portion O_i of the task T_i is 1-task-look-ahead substitutable; if the effective interval t_i is greater than the effective interval t_{i+1}, the optional portion O_i of the task T_i is not 1-task-look-ahead substitutable.

17. A method according to claim 16, wherein:

said step (a) comprising the steps of:

determining a spare interval p_i according to the starting time s_{i+1} and the ending time f_i; and

defining the effective interval t_i as the smaller one of the spare interval p_i and the interval for the processing time o_i of the optional portion O_i; and

said step (c) comprising the steps of:

determining a spare interval p_{i+1} according to the starting time s_{i+2} and the ending time f_{i+1}; and

defining the effective interval t_{i+1} as the smaller one of the spare interval p_{i+1} and the interval for the processing time o_{i+1} of the optional portion O_{i+1}.

18. In a real-time scheduling mechanism for scheduling a plurality of tasks, a method for k-tasks-look-ahead substitutable checking, where k is an integer greater than one, wherein the plurality of tasks T include respective mandatory portions M and optional portions O, the mandatory portions M of the plurality of tasks T are scheduled in a reservation list according to a model so that each of the mandatory portions M_i scheduled in the reservation list has a starting time s_i and an ending time f_i, and after the mandatory portion M_i of each of the tasks T_i is completed, the method is performed on the task T_i, wherein, for each of the tasks T_i, the mandatory portion M_i

has a processing time m_i, the optional portion O_i has a processing time o_i, and the subscript *i* indexes a sequence of starting time for the mandatory portions M scheduled in the reservation list, the method comprising the steps of:

(a) determining an effective interval t_i according to the starting time s_{i+1} of the mandatory portion M_{i+1} of a task T_{i+1}, the ending time f_i of the mandatory portion M_i of the task T_i, and an interval for the processing time o_i of the optional portion O_i of the task T_i;

(b) setting A to be one;

(c) moving the mandatory portion M_{i+A} of a task T_{i+A} to a location in the reservation list by setting the starting time s_{i+A} of the mandatory portion M_{i+A} to be the ending time f_{i+A-1} of the mandatory portion M_{i+A-1} of a task T_{i+A-1} and by changing the ending time f_{i+A} of the mandatory portion M_{i+A} correspondingly;

(d) determining an effective interval t_{i+A} according to the starting time s_{i+A-1} of the mandatory portion M_{i+A-1} of a task T_{i+A-1}, the ending time f_{i+A-1} of the mandatory portion M_{i+A-1}, and an interval for the processing time o_{i+A} of the optional portion O_{i+A} of the task T_{i+A};

(e) if A is less than k, incrementing A by one and proceeding to said step (c);

(f) determining an effective interval *r* by using the effective intervals t_{i+1} to t_{i+k}; and

(g) comparing the effective interval t_i with the effective interval *r*; if the effective interval t_i is less than the effective interval *r*, the optional portion O_i of the task T_i is k-tasks-look-ahead substitutable; if the effective interval t_i is greater than the effective interval *r*, the optional portion O_i of the task T_i is not k-tasks-look-ahead substitutable.

19. A method according to claim 18, wherein said step (a) comprising the steps of:

determining a spare interval p_i according to the starting time s_{i+1} and the ending time f_i; and

defining the effective interval t_i as the smaller one of the spare interval p_i and the interval for the processing time o_i of the optional portion O_i; and

said step (d) comprising the steps of:

determining a spare interval p_{i+A} according to the starting time s_{i+A-1} and the ending time f_{i+A}; and

defining the effective interval t_{i+A} as the smaller one of the spare interval p_{i+A} and the interval for the processing time o_{i+A} of the optional portion O_{i+A}.

20. A method according to claim 18, wherein in said step (f) the effective interval *r* is the interval of maximum length among the effective intervals t_{i+1} to t_{i+k}.

21. In a real-time scheduling mechanism for scheduling a plurality of tasks, a

method for k-tasks-look-ahead substitutable checking, where k is an integer greater than one, wherein mandatory portions M of the plurality of tasks T are scheduled in a reservation list according to a model so that each of the mandatory portions M_i scheduled in the reservation list has a starting time s_i and an ending time f_i, wherein each of the tasks T_i further has an optional portion O_i with processing time o_i, and after the mandatory portion M_i of each of the tasks T_i is completed, the method is performed on the task T_i, where *i* indexes a sequence of starting time for the mandatory portions M scheduled in the reservation list, the method comprising the steps of:

(a) determining an effective interval t_i according to the starting time s_{i+1} of the mandatory portion M_{i+1} of a task T_{i+1}, the ending time f_i of the mandatory portion M_i of the task T_i, and an interval for the processing time o_i of the optional portion O_i of the task T_i;

(b) setting A to be one;

(c) changing the location of the mandatory portion M_{i+A} of a task T_{i+A} in the reservation list by setting the starting time s_{i+A} of the mandatory portion M_{i+A} to be the ending time f_{i+A-1} of the mandatory portion M_{i+A-1} of a task T_{i+A-1} and by changing the ending time f_{i+A} of the mandatory portion M_{i+A} correspondingly;

(d) determining an effective interval t_{i+A} according to the starting time s_{i+A-1} of the mandatory portion M_{i+A-1} of a task T_{i+A-1}, the ending time f_{i+A-1} of the mandatory portion M_{i+A-1}, and an interval for the processing time o_{i+A} of the optional portion O_{i+A}

of the task T_{i+A};

(e) comparing the effective interval t_i with the effective interval t_{i+A}; if the effective interval t_i is less than the effective interval t_{i+A}, the optional portion O_i of the task T_i is k-tasks-look-ahead substitutable; if the effective interval t_i is greater than the effective interval t_{i+A}, the optional portion O_i of the task T_i is not k-tasks-look-ahead substitutable.

(f) ending the method if the optional portion O_i of the task T_i is k-tasks-look-ahead substitutable; and

(g) if A is less than k, incrementing A by one and proceeding to said step (c);

22. A method according to claim 21, wherein said step (a) comprising the steps of:

determining a spare interval p_i according to the starting time s_{i+1} and the ending time f_i; and

defining the effective interval t_i as the smaller one of the spare interval p_i and an interval for the processing time o_i of the optional portion O_i; and

said step (d) comprising the steps of:

determining a spare interval p_{i+A} according to the starting time s_{i+A-1} and the ending time f_{i+A}; and

defining the effective interval t_{i+A} as the smaller one of the spare interval p_{i+A} and the interval for the processing time o_{i+A} of the optional portion O_{i+A}.

23. In an on-line scheduler for scheduling n tasks, T_1 to T_n, with a reservation list, a real-time scheduling method, wherein the reservation list has at most *u* tasks to be put into, each of the tasks T_h has a mandatory portion M_h, an optional portion O_h, a mandatory portion processing time m_h, and a deadline d_h, where *n* is an integer greater than one and *h* is an integer not less than zero and not greater than *n*, the real-time scheduling method comprising:

a1. determining whether there are any mandatory portions in the reservation list; if not, proceeding to step d3;

a2. determining whether there is a mandatory portion waiting to be put into the reservation list; if so, putting the mandatory portion into the reservation list according to an imprecise computation model so that the reservation list has *p* mandatory portions, wherein *p* is an integer not greater than *u*, and the *p* mandatory portions have respective starting times denoted by s with subscripts identical to subscripts of the *p* mandatory portions;

b1. determining whether there is a mandatory portion M_i in the reservation list is scheduled to be executed immediately, where *i* is not less than one and not greater than *p*; if so, proceeding to step b3;

b2. selecting a mandatory portion M_i from the *p* mandatory portions M in the reservation list according to the starting times of the mandatory portions M, and

updating the starting time s_i of the mandatory portion M_i;

b3. starting to execute the mandatory portion M_i and executing said step a2 until the mandatory portion M_i is completed;

c1. determining whether the optional portion O_i of the task T_i is k-tasks-look-ahead substitutable; if not, proceeding to step c4;

c2. removing the task T_i from the on-line scheduler;

c3. proceeding to said step a1;

c4. starting to execute the optional portion O_i of the task T_i, executing said step a2, and then executing step c8 after the execution of said step a2;

c5. determining whether a mandatory portion M_j is to be executed according to the reservation list, where *j* is an integer not less than one and not greater than *p* and not equal to *i*; if so, proceeding to step c8;

c6. determining whether the task T_i is completed; if so, proceeding to step c8;

c7. determining whether the deadline d_i of the task T_i is reached; if not, proceeding with said step c4;

c8. removing the task T_i from the on-line scheduler;

d1. determining whether there is a task in the on-line scheduler; if not,

proceeding to step d3;

d2. proceeding to said step a1; and

d3. ending the method.

24. A real-time scheduling method according to claim 23, wherein, in said step b1, the starting time s_i of the mandatory portion M_i is the minimum among the values of starting time associated with the *p* mandatory portions.

25. In an on-line scheduler for scheduling *n* tasks, T_1 to T_n, with a reservation list, a real-time scheduling method, wherein the reservation list has at most *u* tasks to be scheduled in, each of the tasks T_h has a mandatory portion M_h, an optional portion O_h, a mandatory portion processing time m_h, and a deadline d_h, where n is an integer greater than one and *h* is an integer not less than zero and not greater than *n*, the real-time scheduling method comprising:

a1. determining whether there are any mandatory portions in the reservation list; if not, proceeding to step d3;

a2. determining whether there is a mandatory portion waiting to be put into the reservation list; if so, putting the mandatory portion into the reservation list according to an imprecise computation model so that the reservation list has *p* mandatory portions, wherein each of the *p* mandatory portions M_k has a starting time s_k and an ending time e_k, where *p* is an integer not greater than *u* and *k* is an integer not greater than *u*;

b1. determining whether there is a mandatory portion M_i in the reservation list is scheduled to be executed immediately, where *i* is not less than one and not greater than *p*; if so, proceeding to step b3;

b2. selecting a mandatory portion M_i from the *p* mandatory portions M in the reservation list according to the starting times of the mandatory portions M, and updating the starting time s_i of the mandatory portion M_i;

b3. starting to execute the mandatory portion M_i and executing said step a2 until the mandatory portion M_i is completed;

c1. determining whether the optional portion O_i of the task T_i is k-tasks-look-ahead substitutable; if not, proceeding to step c12;

c2. selecting a mandatory portion M_q from the reservation list, wherein a starting time s_q of the mandatory portion M_q is a minimum among the starting times which are greater than the ending time e_i of the mandatory portion M_i, and q is an integer not less than one and not greater than *p*;

c3. determining a spare interval g_i, wherein the spare interval g_i is defined by the starting time s_q and the ending time e_i and has a length defined by the difference between the starting time s_q and the ending time e_i;

c4. determining an insertion time v_i, wherein the insertion time v_i is defined by the difference between the deadline d_i and the length of the spare interval g_i;

c5. determining whether the insertion time v_i is greater than the ending time e_i and less than the deadline d_i; if so, proceeding to step c8;

c6. removing the task T_i from the on-line scheduler;

c7. proceeding to said step a1;

c8. starting to execute the mandatory portion M_q and executing said step a2 until the mandatory portion M_q is completed;

c9. starting to execute the optional portion O_i of the task T_i, executing said step a2, and then executing step c11 after the execution of said step a2;

c10. determining whether a mandatory portion M_j is to be executed according to the reservation list, where *j* is an integer not less than one and not greater than *p* and not equal to *i*; if so, proceeding to step c13;

c11. determining whether the task T_i is completed; if so, proceeding to step c13;

c12. determining whether the deadline d_i of the task T_i is reached; if not, proceeding to said step c9;

c13. removing the task T_i from the on-line scheduler;

d1. determining whether there are any tasks in the on-line scheduler; if not, proceeding to step d3;

d2. proceeding to said step a1; and

d3. ending the method.

26. A real-time scheduling method according to claim 25, wherein, in said step b1, the starting time s_i of the mandatory portion M_i is the minimum among the starting times associated with the *p* mandatory portions M.

27. In an on-line scheduler for scheduling *n* tasks, T_1 to T_n, with a reservation list, a real-time scheduling method, wherein the reservation list has at most *u* tasks to be put into, each of the tasks T_h has a mandatory portion M_h, an optional portion O_h, a mandatory portion processing time m_h, and a deadline d_h, where n is an integer greater than one and *h* is an integer not less than one and not greater than *n*, the real-time scheduling method comprising:

a1. determining whether there are any mandatory portions in the reservation list; if not, proceeding to step d3;

a2. determining whether there is a mandatory portion waiting to be put into the reservation list; if so, putting the mandatory portion M into the reservation list according to an imprecise computation model so that the reservation list has *p* mandatory portions, wherein *p* is an integer greater than *r* and not greater than *u*, and each of the *p* mandatory portions M_k has a starting time s_k and an ending time e_k, where *p* is an integer not greater than *u* and *k* is an integer not greater than *u*;

b1. determining whether there is a the mandatory portion M_i in the reservation

list is scheduled to be executed immediately, where *i* is not less than one and not greater than *p*; if so, proceeding to step b3;

b2. selecting a mandatory portion M_i from the *p* mandatory portions in the reservation list according to the starting times of the mandatory portions M, and updating the starting time s_i of the mandatory portion M_i;

b3. starting to execute the mandatory portion M_i and executing said step a2 until the mandatory portion M_i is completed;

c1. determining whether the optional portion O_i of the task T_i is k-tasks-look-ahead substitutable; if not, proceeding to step c7;

c2. determining a block separation number b_i, wherein the block separation number b_i is one plus the number of a mandatory portion set in the reservation list;

c3. selecting a mandatory portion M_q from the reservation list, wherein the mandatory portion M_q belongs to the mandatory portion set, and the starting time s_q of the mandatory portion M_q is a minimum among starting times associated with the mandatory portion set and is greater than the ending time e_i, and subscript *q* is an integer not less than one and not greater than *p*;

c4. determining a spare interval g_i, wherein the spare interval g_i is defined by the starting time s_q and the ending time e_i and has a length defined by the difference between the starting time s_q and the ending time e_i;

c5. determining an optional-portion processing period o_t, wherein the optional-portion processing period o_t is defined by the length of the spare interval g_i divided by the block separation number b_i;

c6. starting to execute the optional portion O_i of the task T_i for the optional-portion processing period o_t, and executing said step a2 until the optional portion O_i of the task T_i has been executed for optional-portion processing period o_t;

c7. removing the task T_i from on-line scheduler;

c8. starting to execute a task T_r associated with one of the mandatory portion set beginning from the mandatory portion M_q according to the starting times of the mandatory portions in the mandatory portion set, until all of the mandatory portions in the mandatory portion set are executed and all optional portions associated with the mandatory portions in the mandatory portion set are executed for the optional-portion processing period o_t;

d1. determining whether there are any tasks in the on-line scheduler; if not, proceeding to step d3;

d2. proceeding to said step a1; and

d3. ending the method.

28. A real-time scheduling method according to claim 27, wherein, in said step b1, the starting time s_i of the mandatory portion M_i is the minimum among the starting

times of the *p* mandatory portions.

29. A real-time scheduling method according to claim 27, wherein, in said step c2, the optional portion set is a block of consecutive mandatory portions including the mandatory portion M_q.

30. A real-time scheduling method according to claim 27, wherein, in said step c2, the block separation number b_i is set to two when the mandatory portion M_q is not immediately followed by another mandatory portion in the reservation list and the mandatory portion set contains the mandatory portion M_q only.

31. A real-time scheduling method according to claim 27, wherein said step c8 comprises the steps of:

executing a mandatory portion M_r of the task T_r;

starting to execute an optional portion O_r of the task T_r until the optional portion O_r of the task T_r has been executed for the optional-portion processing period o_t;

removing the task T_r from the on-line scheduler; and

proceeding to said step of executing a mandatory portion M_r of the task T_r for a mandatory portion with the earliest starting time among the mandatory portions of the mandatory portion set that have not been executed, until the mandatory portions of the mandatory portion set are executed.

32. A method for quality-of-service-controllable real-time scheduling, the method comprising the steps of:

(a) regulating the number of input tasks which are to be forwarded to an on-line scheduling unit through a regulating unit;

(b) by the on-line scheduling unit, selecting a real-time scheduling method, scheduling tasks forwarded to the on-line scheduling unit according to the real-time scheduling method, and outputting a scheduling result; and

(c) by an evaluating unit, evaluating the scheduling result, feeding a first set of parameters into the regulating unit for a coarse adjustment, and feeding a second set of parameters into the on-line scheduler for a fine adjustment.

33. A method according to claim 32, wherein the regulating unit, the on-line scheduling unit, and the evaluating unit are implemented as respective software routines.

34. A method according to claim 32, wherein each of the tasks includes a mandatory portion and an optional portion, and the coarse adjustment is performed by controlling a proportion of the optional portion to be executed.

35. A method according to claim 32 or 34, wherein said step (a) is performed by the regulating unit according to the first set of parameters.

36. A method according to claim 32 or 34, wherein, in said step (b), the

real-time scheduling method is selected according to the second set of parameters.

37. A method according to claim 36, wherein the real-time scheduling method is a scheduling method in which the mandatory portions of the input tasks are executed as soon as possible and the optional portions of the inputted tasks are substitutable (MOS method).

38. A method according to claim 36, wherein the real-time scheduling method is a scheduling method in which the mandatory portions of the input tasks are executed as soon as possible and the substitutable optional portions of the inputted tasks are postponed (MOP method).

39. A method according to claim 36, wherein the real-time scheduling method is a scheduling method in which the mandatory portions of the input tasks are executed as soon as possible and the optional portions of the inputted tasks are to be executed fairly (MOF method).

40. A method according to claim 32 or 34, wherein, in said step (c), the evaluating unit evaluates the scheduling result according an evaluation standard, and the evaluation standard includes a task rejection rate.

41. A method according to claim 32 or 34, wherein, in said step (c), the evaluating unit evaluates the scheduling result according an evaluation standard, and the evaluation standard includes a task suspend/discard rate.

42. A method according to claim 32 or 34, wherein, in said step (c), the

evaluating unit evaluates the scheduling result according an evaluation standard, and the evaluation standard includes an idle rate.

43. A method according to claim 32 or 34, wherein, in said step (c), the evaluating unit evaluates the scheduling result according an evaluation standard, and the evaluation standard includes a slack time.

44. A method according to claim 32 or 34, wherein the first set of parameters includes a token generation rate.

45. A method according to claim 32 or 34, wherein the first set of parameters includes a token number.

46. A method according to claim 32 or 34, wherein the first set of parameters includes a queue length.

47. A method according to claim 32 or 34, wherein the second set of parameters includes a real-time scheduling selection parameter.

48. A method according to claim 32 or 34, wherein the second set of parameters includes a substitutable check parameter.

49. A method according to claim 32 or 34, wherein the second set of parameters includes a parameter indicative of a maximum allowable execution proportion of the optional portion.

* * * * *